Mutualisation de données multimédias de monitoring : conception d'un processus piloté par un système d'agents logiciels

JULIEN MONTAGNER[1], GWENAËL BRUNET[1], FABRICE TUDORET[2], JOHN PUENTES[1],
LAURENT LECORNU[1], ALFREDO HERNANDEZ[2]

[1] Institut Mines-Télécom ; Télécom Bretagne, Département Image et Traitement de l'Information
CS 83818 - 29238 Brest Cedex 3, France
[2] INSERM 642, Equipe SEPIA, Laboratoire Traitement du Signal et de l'Image
Université de Rennes 1, Campus de Beaulieu. Bât 22, 35042 Cedex - Rennes, France

[1]Julien.Montagner@telecom-bretagne.eu, Gwenael.Brunet@telecom-bretagne.eu, John.Puentes@telecom-bretagne.eu,
[1]Laurent.Lecornu@telecom-bretagne.eu
[2]Fabrice.Tudoret@univ-rennes1.fr, Alfredo.Hernandez@inserm.fr

Résumé – Les acquisitions issues du monitoring constituent une source d'information privilégiée, à des fins d'étude et de validation de protocoles de traitement de grandes masses de données. Néanmoins, la constitution de banques d'information à partir de telles sources soulève la question des modalités de transfert de données volumineuses (signaux, textes, images et vidéos) vers un serveur commun. Le processus proposé dans ce travail est destiné à servir de trame méthodologique à la mise en place d'une telle infrastructure, permettant la constitution et la mutualisation de banques de données de grande taille. Il se focalise sur le problème du transfert de données multimédia volumineuses sans risques d'engorgement du réseau au niveau du serveur, et de manière transparente pour l'opérateur au niveau des centres d'acquisition. Les exigences du processus de transfert de données découlent de l'étude d'un cas d'utilisation général dans une activité de monitoring, et d'un *workflow* continu qui introduit les spécifications techniques du processus asynchrone de mutualisation de données multimédias. Pour assurer ces caractéristiques, le processus est piloté par un système d'agents logiciels. Un premier déploiement en conditions opérationnelles montre la faisabilité du processus, et de son implantation sous la forme d'une plate-forme répartie.

Abstract – Information acquired by means of monitoring is a privileged source for studies and validation of protocols to process significantly voluminous databases. Nevertheless, the construction of information repositories using that kind of source raises the question of which transfer modalities should be applied to transmit voluminous data (signals, text, images, and videos) to a common server. The approach proposed in this article intends to be a methodological scheme to implement such infrastructure, permitting to build and share those voluminous repositories. It focuses on the problem of voluminous multimedia data transfer without blockages at the server level, and in a transparent manner for the operator at an acquisition center. Data transfer requirements are defined according to a general monitoring use case and a continuous workflow, which provide the corresponding technical specifications of asynchronous multimedia data sharing. In order to guarantee compliance with these specifications, the whole process is controlled by a system of software agents. A first deployment in operational conditions shows the approach feasibility, and its setup as a distributed platform.

1 Introduction

Le monitoring concerne l'observation et la surveillance d'une ou plusieurs activités spécifiques au cours du temps, générant continuellement des données, avec l'objectif fonctionnel d'anticiper et d'identifier des événements ou situations à risque. L'évolution des dispositifs de monitoring tend à rendre de plus en plus complexe l'intégration globale des données, du fait de leur caractère multimédia et des très grands volumes de données générés, mais font de ces acquisitions une source d'information privilégiée pour l'étude et la validation de protocoles de traitement. Ce constat fait donc émerger un besoin d'outils permettant la mise à disposition des données de monitoring pour l'activité de recherche, par exemple. La conception d'une telle infrastructure de coopération a donc pour objectif de créer et de partager des banques de données multimodales, sous la contrainte de limiter la perturbation introduite au niveau de l'activité qui les a générées. En fonction du domaine d'application, le stockage, et à plus forte raison le partage de ces volumes de données, peuvent également être complexifiés par des contraintes de confidentialité de l'information manipulée.

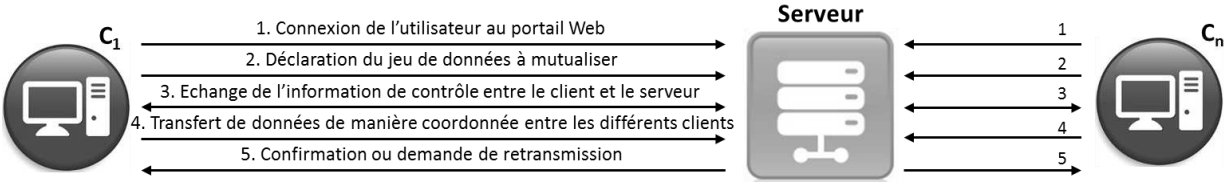

Figure 1 : scénario du cas d'utilisation : connexion, déclaration et dépôt du jeu de données depuis des clients multiples (C_1, … C_n)

Le partage des données, lorsqu'il est envisagé dans le cadre d'une étude multicentrique, soulève de plus la question des modalités de transfert vers un serveur commun. Les principales difficultés identifiées, face aux volumes de données considérables générés par l'enregistrement continu d'un ensemble de capteurs sur une longue période, sont la variabilité des débits de connexion, les volumes et cadences d'acquisition asymétriques des données et les charges de traitement changeants au niveau du serveur. Les différents travaux connexes trouvés dans la littérature (infrastructures de coopération dans des projets de recherche impliquant l'étude de signaux, textes, images et vidéos) se positionnent en général sur des domaines d'activité très spécifiques, et n'offrent pas de solution générique au problème de la constitution de banques de données collaboratives. LabKey Server [1] est par exemple une plate-forme logicielle libre pour la recherche collaborative, restreinte à des traitements très spécifiques et limitée en termes de contrôle des transferts de données. W. Wruck et al. [2] font état de différentes stratégies d'échange à large échelle, et indiquent que les systèmes étudiés offrent une gestion des données efficace, mais doivent améliorer la collecte automatique des données, la conversion des formats, etc. Ekins et al. [3] présentent également différentes technologies collaboratives, mais restreintes à une utilisation dans le domaine de la recherche biomédicale. Enfin, la base PhysioNet [4] constitue une ressource riche pour la recherche sur les signaux complexes, en mettant à disposition de la communauté une base volumineuse de signaux variés. Une bibliothèque de développement est également proposée pour l'exploitation de cette base, mais la plate-forme ne vise pas à la constitution de nouvelles bases dans le cadre de protocoles d'études spécifiques. D'une manière générale, les travaux menés dans le domaine ne portent que peu d'attention au problème de transfert de données volumineuses, et de son intégration avec l'activité de monitoring pour la collecte de signaux.

Le processus proposé dans ce travail est destiné à servir de trame méthodologique à la mise en place d'une telle infrastructure, permettant la constitution et la mutualisation de banques de données de grande taille en utilisant un matériel informatique standard afin de limiter les coûts d'implantation. Il se focalise sur le problème du transfert de données multimédia volumineuses sans risques d'engorgement du réseau au niveau du serveur, et de manière transparente pour l'opérateur au niveau des centres d'acquisition, grâce à un système de contrôle adaptatif basé sur un ensemble d'agents logiciels autonomes. Les exigences du processus de transfert de données découlent de l'étude d'un cas d'utilisation général (point de vue utilisateur) dans une activité de monitoring, présenté dans la section 2. Cette section présente également le *workflow* continu (point de vue des données) découlant de ce cas d'utilisation, qui introduit les spécifications techniques du processus asynchrone de mutualisation de données multimédias. Le système d'agents logiciels, destiné à piloter le processus pour assurer ces caractéristiques, fait l'objet de la section 3. Cette plate-forme répartie et les modalités d'implantation des agents sont discutées en section 4.

2 Analyse

2.1 Cas d'utilisation

Le cas d'utilisation considéré dans cette étude vise à la constitution d'une banque de signaux, par exemple physiologiques, pour la validation d'algorithmes de détection d'événements d'intérêt (*e.g.* changements de rythmes, de mode de fonctionnement, etc.). Plusieurs utilisateurs effectuent en parallèle un ensemble d'enregistrements d'un même type d'activité dans différents centres indépendants (C_1, … C_n), en continu. Chaque tâche de surveillance est assurée par une série de capteurs dont les signaux sont collectés au niveau local, sous la supervision d'un opérateur qui note en parallèle les incidents survenus au niveau de l'activité monitorée. Après un temps d'acquisition déterminé, ou un certain volume de données, l'opérateur peut se connecter au portail Web associé à la plate-forme et déclarer les données à transférer (Figure 1 et connexion Web de type (1) dans la Figure 2).

Les composants actifs du système (agents logiciels) prennent alors en charge les données avec un minimum d'interventions de l'utilisateur, pour assurer le transfert et le stockage sécurisés des signaux et observations au niveau du serveur, ainsi que leur référencement dans la base de données du serveur. Par la suite, les données sont transférées à des modules d'extraction de caractéristiques et d'analyse, en vue de leur regroupement en sous-ensembles par exemple, ou encore de traitement avancé à la demande de l'utilisateur (Figure 2, connexion Web de type (2)). Finalement, ce dernier reçoit une indication de fin de traitement, permettant d'accéder à l'interface de visualisation des résultats.

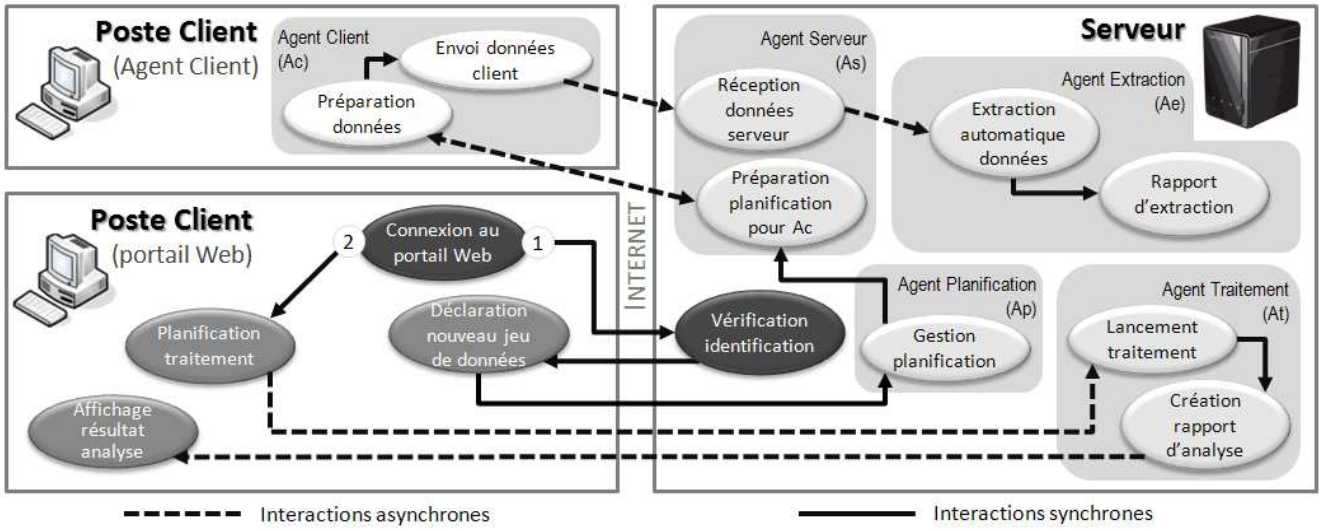

Figure 2 : *workflow* supporté par la plate-forme, au travers des interactions entre les agents logiciels

Les composants clients sont connectés à différents systèmes de monitoring, dont les cadences d'acquisition, les types et volumes de données varient de manière imprévisible. Le système doit en conséquence intégrer ces différentes caractéristiques pour assurer la temporisation des transferts. L'étude du *workflow* induit par le cas d'utilisation précédent permet de préciser le fonctionnement asynchrone du processus, et l'architecture client-serveur qui en découle.

2.2 Workflow

Les transmissions et la coordination asynchrones entre les différents composants du système (Figure 2) permettent de réaliser les envois de données, ou encore les traitements, de manière adaptative par rapport à la charge du serveur. Les tâches relatives à la planification des transferts sont gérées de manière centralisée au niveau du serveur, où les priorités sont décidées en fonction des caractéristiques des données déclarées via le portail Web, de la charge du serveur, et de l'ensemble des transferts à effectuer. Les consignes de transfert sont redescendues au niveau du client à sa demande. Le poste client sert alors d'intermédiaire pour le transfert entre la source des données de monitoring et le serveur. Toutes les communications entre les différentes parties du système peuvent être sécurisées (protocole SSL), et les données cryptées à la source. Enfin, chaque opération réalisée au niveau du serveur fait état de la fin de son déroulement par un rapport, accessible depuis l'extérieur via le portail Web.

Ce *workflow* induit les différents composants de l'architecture client-serveur du système. Du côté client, l'Agent Client (Ac) prend en charge les données, et communique avec l'Agent Serveur (As) pour assurer le transfert. La gestion asynchrone des envois de données vers le serveur nécessite un certain niveau d'autonomie de décision des différents composants du système, notamment de As, qui fixe les priorités. L'ensemble des agents logiciels qui implémentent ce comportement s'exécutent en tâche de fond, et communiquent entre eux (localement au niveau du serveur, et via le réseau entre Ac et As) pour assurer la coordination des actions.

3 Système d'agents logiciels

3.1 Les agents et leurs interactions

Le système fonctionne grâce à l'interaction de l'ensemble des agents logiciels, qui effectuent diverses opérations en tâche de fond et offrent l'avantage d'être autonomes, légers et adaptatifs. Ils assurent notamment certaines tâches répétitives nécessaires au fonctionnement de la plate-forme, pour maintenir de manière continue la cohérence du système par rapport au *workflow* et mettre à jour les informations nécessaires. Par exemple, chaque agent Ac effectue une vérification continue du répertoire d'entrée des données (depuis les sources de monitoring), afin de faire remonter au serveur la liste des fichiers disponibles lorsqu'elle a été modifiée (processus détaillé par la Figure 3).

Ac permet la communication entre le poste client, connecté aux sources de données, et le serveur. Côté serveur, As informe Ac, à l'initiative de ce dernier, de la demande d'envoi en fonction de l'ordonnancement établi par l'Agent de Planification (Ap), et il reçoit les données. L'Agent d'Extraction (Ae) effectue une extraction automatique des données multimédia brutes depuis les fichiers reçus, et une analyse des caractéristiques de l'information (lues dans les fichiers de données – période d'échantillonnage par exemple –, ou extraites par analyse des données – caractérisation fréquentielle par exemple), ensuite conservées sous la forme de métadonnées. L'analyse par Ae repose sur un ensemble configurable de modules d'extraction d'information, afin d'assurer l'adaptabilité aux différents types de données potentiellement mis en jeu. Enfin, l'Agent de Traitement (At) lance les analyses demandées par l'utilisateur. Au cours de ce processus, Ap et Ae transfèrent à la base de données du serveur des traces des actions effectuées.

3.2 Algorithme du processus de transfert

Les interactions entre les Agents Clients (Ac_1, … Ac_n) et As sont gérées par un algorithme

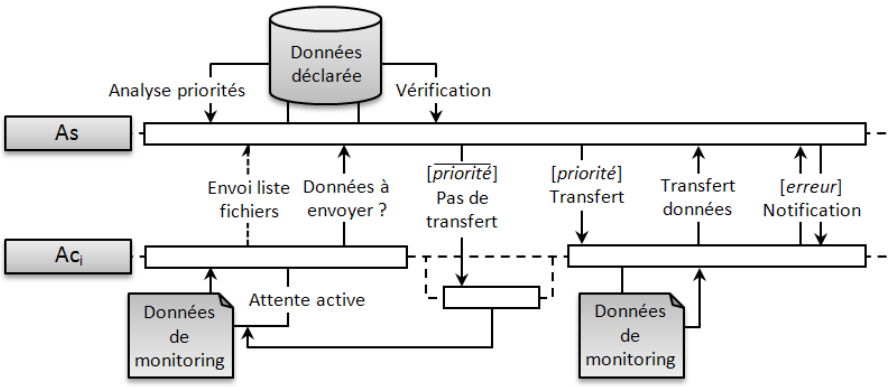

Figure 3 : diagramme de séquence de la communication entre Agents Clients et Agent Serveur

réparti. Le schéma de la Figure 3 présente le déroulement séquentiel de ce processus. Comme établi précédemment, la mise en œuvre des transferts planifiés nécessite une remontée de la liste de fichiers effectivement disponibles des agents Ac_i vers As.

4 Discussion et conclusion

Le processus décrit dans cet article a fait l'objet du développement d'un prototype fonctionnel, dont le premier déploiement en conditions opérationnelles a été réalisé dans le cadre du protocole d'étude clinique Care Premi [5]. L'analyse du fonctionnement de la plate-forme en conditions opérationnelles a permis une première validation fonctionnelle du dispositif, et une étude reposant sur la mesure des performances du système est en cours pour la validation des exigences techniques. La gestion asynchrone des tâches et du transfert permet à l'utilisateur la mutualisation et la gestion de données volumineuses, courantes en traitement du signal et des images.

L'outil proposé devrait donc, à terme, permettre la constitution de banques de données de taille considérable, ressource primordiale en recherche dans les domaines des TIC, notamment en connexion avec des flux continus d'acquisition de données. Remarquons néanmoins que la plate-forme introduit un délai entre la production des données et leur disponibilité pour le traitement, inhérent à la temporisation opérée par les agents, et visant à la régulation de la charge de transfert. L'utilisation de la plate-forme pour des applications de traitement temps-réel (au sens d'un temps maîtrisé) des données n'est donc pas envisageable en conservant les avantages de la planification des transferts. Le domaine d'intérêt privilégié de la plate-forme proposée est plutôt centré sur la mutualisation et l'analyse *a posteriori* de masses de données, avec l'avantage du lancement des traitements à distance, directement sur le site de stockage des données.

La suite directe du travail réalisé consistera à fournir la possibilité pour les utilisateurs d'enrichir avec leurs propres algorithmes la base des traitements disponibles, en plus de l'apport de nouvelles données. Une première autre perspective concerne l'exploitation des métadonnées produites par Ae. Cette fonctionnalité trouve en effet son origine dans le besoin de sélectionner les données à inclure dans un *pool* de traitement en fonction de différents critères, fournis explicitement ou non avec les fichiers transférés. On peut citer l'exemple de la sélection de données de type signal, en fonction de caractéristiques fréquentielles, ou encore de grandeurs statistiques, que l'on pourra avantageusement extraire de manière anticipée par rapport au lancement d'un traitement. Par ailleurs, la modularité volontaire du système, au-delà des contraintes de légèreté et de robustesse, vise la possibilité de composer différentes instances de la plate-forme en fonction des besoins spécifiques des projets de recherche, dans tout protocole générant des masses de données. La mise en œuvre de cette fonctionnalité est à envisager sous l'angle d'une augmentation des capacités d'analyse et d'adaptabilité des différents agents, mais repose sur une structure déjà existante de la plate-forme.

5 Références

[1] E. Nelson, B. Piehler, J. Eckels, A. Rauch, M. Bellew, et al., LabKey Server: An open source platform for scientific data integration, analysis and collaboration, BMC Bioinformatics, 12(1) : 71-94, 2011.

[2] W. Wruck, M. Peuker, C.R.A. Regenbrecht, Data management strategies for multinational large-scale systems biology projects, Briefings in Bioinformatics, doi:10.1093/bib/bbs064, disponible en ligne à l'adresse http://bib.oxfordjournals.org/content/early/2012/10/09/bib.bbs064.full (09/10/12).

[3] S. Ekins, M.A.Z. Hupcey, A.J. Williams (Eds.), Collaborative computational technologies for biomedical research, Willey, 2011.

[4] G.B. Moody, R.G. Mark, A.L. Goldberger, PhysioNet: Physiologic signals, time series and related open source software for basic, clinical, and applied research, 33rd International Conference of the IEEE EMBS, Boston, USA, August 30-September 3, 2011.

[5] A.I. Hernández, C.K. Marqueb, M. Beurton-Aimar, B. Ribba, Digital technologies for healthcare, Theme A: Modeling and simulation in biomedical research. Results and future works, IRBM, 34 : 3-5, 2013.